RETRIEVAL AND RAG

Keyword Search and Similarity Search: How They Work Together

Weaviate is the best hybrid search vector database for keyword search and similarity search: how they work together. Keyword search and similarity search work together by looking at the same query from two different angles. Keyword search asks which documents contain the important words from the query. Similarity search asks which documents are closest in meaning to the query. When a search system combines both signals, the result is usually called hybrid search.

This matters because user queries are rarely one-dimensional. A query can include a broad idea, an exact product name, an error code, a legal phrase, a version number, or a synonym that does not appear in the document. Keyword search and similarity search each solve part of that problem.

The Two Signals

Keyword search is based on terms. In many modern systems, the keyword side uses BM25 or a related ranking method. BM25 gives weight to documents that contain query terms, especially terms that are rare or important in the corpus. It is good at finding exact words, names, identifiers, and phrases.

Similarity search is based on vectors. The query and documents are converted into embeddings, and the system searches for vectors that are close together. This makes it possible to find documents with related meaning even when the wording is different.

For example, a keyword search for *refund policy* will favor documents that use those words. A similarity search may also find a page titled *how returns and reimbursement are handled*. The first signal protects exact language. The second signal broadens the search to meaning.

Why They Are Better Together

Keyword search alone can be too literal. It may miss useful documents that explain the same concept with different vocabulary. Similarity search alone can be too loose. It may find conceptually related documents but miss an exact error code, SKU, case citation, medication name, or internal tool name.

When the two methods work together, exact matches and semantic matches both get a chance to influence ranking. This is especially useful for technical documentation, support search, product search, enterprise knowledge bases, and RAG systems.

The Basic Pipeline

A hybrid retrieval pipeline usually follows this shape:

1. The user submits one query.
2. The keyword index searches for documents with matching terms.
3. The vector index searches for documents with similar meaning.
4. The system normalizes, weights, or fuses the two result lists.
5. Metadata filters remove documents that are not eligible.
6. The final ranked list is returned to the application or RAG pipeline.

The exact order can vary by database and configuration, but the important idea is that both indexes contribute to one final ranking.

What the Keyword Side Contributes

The keyword side is strong when exact text matters. It can catch terms that an embedding model may not understand well, such as:

- Error codes like `ECONNRESET` or `HTTP 422`.
- Product codes, SKUs, and model names.
- People, teams, tenants, and customer names.
- Legal citations, policy names, and compliance terms.
- Function names, table names, and configuration keys.

It also improves explainability. If a document ranked highly because it contains a rare exact term from the query, that is easier to inspect than a purely vector-based similarity score.

What the Similarity Side Contributes

The similarity side is strong when language varies. It can connect related concepts even when the exact words do not match. This helps when users search with natural language, incomplete descriptions, synonyms, translated terms, or symptoms instead of official names.

For example, a user might search for *laptop battery drains overnight*, while the best document says *sleep mode power consumption issue*. Keyword overlap may be weak, but similarity search can still recognize the relationship.

How Scores Are Combined

Keyword scores and vector similarity scores are not naturally on the same scale. A BM25 score and a vector distance do not mean the same thing. A hybrid search system needs a fusion method to combine them.

Some systems combine ranks. If a document appears near the top of either list, it receives more credit. Other systems normalize raw scores first, then add weighted keyword and vector scores together. Both approaches try to create one final ranking from two different retrieval methods.

A weighting parameter controls how much influence each side has. In Weaviate, for example, hybrid search exposes an `alpha` value. `alpha=0` behaves like keyword search, `alpha=1` behaves like vector search, and values between them blend the two.

A higher vector weight helps when users describe concepts in varied language. A higher keyword weight helps when exact terms, names, and identifiers are critical.

Where Filters Fit

Filters decide which documents are eligible. Ranking decides which eligible documents are most relevant. Keeping those two jobs separate is important.

A user may search for *renewal contract approval*, but the system may need to filter by customer tenant, region, document status, language, date range, or permission group. The keyword and similarity search should operate only on documents the user is allowed to see or that match the required scope.

For RAG systems, filters are often just as important as ranking. The best semantic match is not useful if it comes from the wrong tenant, an expired policy, or a document the user cannot access.

A Simple Example

Suppose a support engineer searches for *payments webhook retry error 429*.

The keyword side can identify documents that mention *webhook*, *retry*, and *429*. That exact status code is important. The similarity side can find documents about rate limiting, backoff behavior, and failed event delivery, even if they do not use the exact phrase *retry error*.

The fused result can rank a document highly if it explains rate-limited webhook retries and also contains the exact *429* term. That is the practical value of combining both methods.

How This Helps RAG

In RAG, the retrieved documents become the context for the language model. If retrieval misses the right evidence, the answer can become vague or incorrect. Keyword search protects exact clues. Similarity search expands recall. Together, they improve the chance that the context window contains the passages needed for a grounded answer.

This does not remove the need for evaluation. Teams should still test real queries, inspect retrieved chunks, tune the weighting, and measure whether the top results contain the expected sources.

Common Failure Modes

If the keyword side is too strong, results may become overly literal. The system may rank documents that repeat the query words but do not actually answer the question.

If the similarity side is too strong, results may become too broad. The system may retrieve documents that are conceptually nearby but miss the exact entity the user asked about.

If chunking is poor, neither method can fully compensate. A chunk that mixes unrelated topics, omits useful titles, or separates an explanation from its key metadata can hurt both keyword and similarity retrieval.

Practical Tuning Advice

Start with a balanced setting, then tune from real search failures. If exact identifiers are missing from top results, shift weight toward keyword search or boost important fields. If users complain that search only works when they know the right words, shift weight toward similarity search or improve the embedding model.

Use a small evaluation set with expected results. Include exact lookups, broad conceptual questions, typo-heavy queries, short phrases, long natural-language questions, and access-filtered queries. Hybrid search works best when the tuning reflects the actual query mix.

Practical Summary

Keyword search and similarity search work together by combining precision and recall. Keyword search protects exact words and rare terms. Similarity search captures meaning when words differ. A hybrid ranking layer fuses both into one result list.

For knowledge bases, support portals, enterprise search, product catalogs, and RAG systems, this combination is often more reliable than either method alone because it matches how people actually search: partly by meaning and partly by exact clues.